

## DISK ARRAY CONTROLLER

**Publication number:** JP2000267815

**Publication date:** 2000-09-29

**Inventor:** TANAKA ATSUSHI; FUJIBAYASHI AKIRA; FUJIMOTO KAZUHISA; KANAI HIROKI; MINOWA NOBUYUKI; MIKAMI HIKARI; ASARI MAKOTO

**Applicant:** HITACHI LTD; HITACHI SOFTWARE ENG; HITACHI VIDEO & INF SYST

**Classification:**

- **international:** G06F3/06; G06F11/20; G06F3/06; G06F11/20; (IPC1-7): G06F3/06; G06F3/06

- **European:** G06F3/06D; G06F11/20L8C

**Application number:** JP19990071400 19990317

**Priority number(s):** JP19990071400 19990317

Also published as:

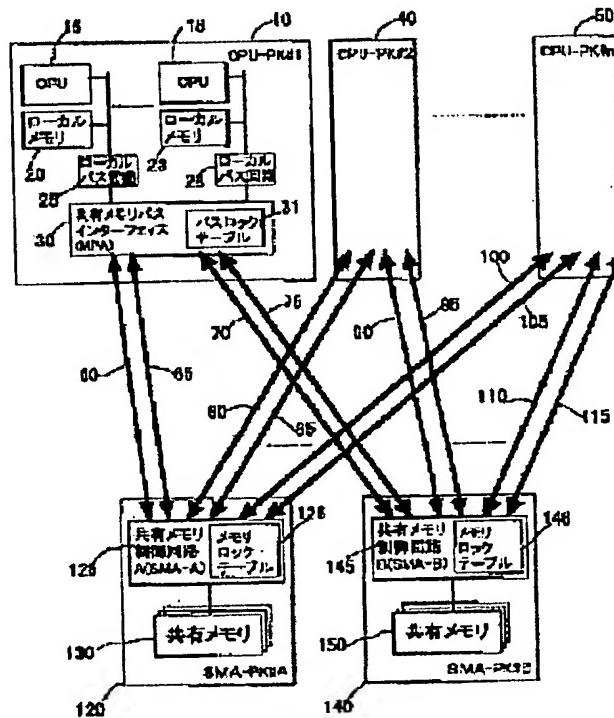
- EP1037137 (A2)
- US6502167 (B1)
- EP1037137 (A3)

[Report a data error here](#)

### Abstract of JP2000267815

**PROBLEM TO BE SOLVED:** To perform efficient path selection without data noncoincidence, etc., to reduce processing wait time and to increase the number of processing transactions by referring to a path lock table showing the use situation of each path in a double write mode.

**SOLUTION:** A double write operation transmits data from a CPU to an MPA 30, selects a 0th path 60 or a 1st path 65 to be a master surface and transmits it to a shared memory control circuit-A125 of an SMA-PK#A120. After finishing write processing, the status of end information is transmitted to the MPA 30 from the circuit-A125 through the same shared memory. After that, a master side path is held by the circuit-A125 and can not be used by the other accesses. A slave side releases a path after finishing transfer by selecting a 0th path 70 or a 1st path 75 to be a slave surface, transmitting data to a shared memory control circuit-B145 and transmitting the status of end information to the MPA 30 from an SMA-PK#B140 by the same shared memory bus.



Data supplied from the esp@cenet database - Worldwide



(19)日本国特許庁 (JP)

## (12) 公開特許公報 (A)

(11)特許出願公開番号

特開2000-267815

(P2000-267815A)

(43)公開日 平成12年9月29日(2000.9.29)

(51)Int.Cl.<sup>7</sup>  
G 0 6 F 3/06識別記号  
3 0 1  
3 0 2  
5 4 0F I  
G 0 6 F 3/06デマコード\*(参考)  
3 0 1 C 5 B 0 6 5  
3 0 2 A  
5 4 0

審査請求 未請求 請求項の数9 ○L (全 26 頁)

(21)出願番号 特願平11-71400

(22)出願日 平成11年3月17日(1999.3.17)

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(71)出願人 000233055

日立ソフトウェアエンジニアリング株式会  
社

神奈川県横浜市中区尾上町6丁目81番地

(71)出願人 000233136

株式会社日立画像情報システム  
神奈川県横浜市戸塚区吉田町292番地

(74)代理人 100068504

弁理士 小川 勝男

最終頁に続く

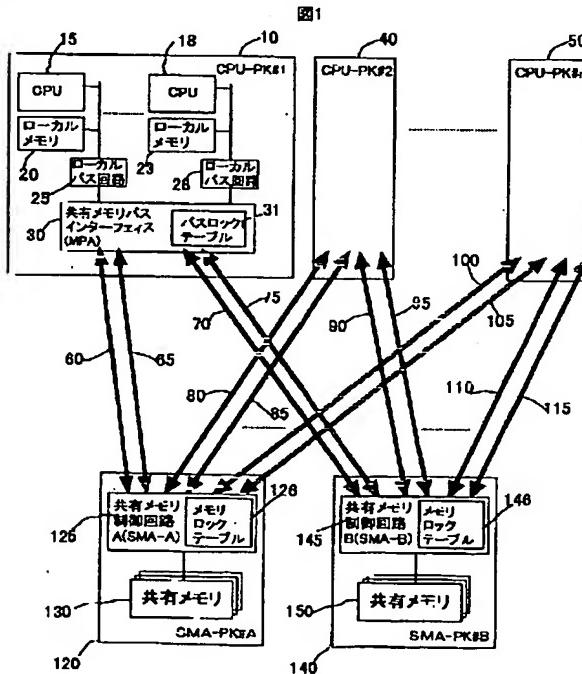
(54)【発明の名称】ディスクアレイ制御装置

(57)【要約】

【課題】ディスクアレイ制御装置において、2重化された共有メモリのメアクセスの応答時間の短縮と、処理トランザクション数の増加にある。

【解決手段】ホストコンピュータ/ディスク装置との複数のインターフェース部と、各インターフェース部とそれぞれアクセスバスにより1対1に接続される2重化された共有メモリ部と、上記複数のインターフェース部に接続されるセレクタと、上記セレクタに接続されるキャッシュメモリとを設ける。上記複数のインターフェース部と上記セレクタとの間のアクセスバスの本数は、上記キャッシュメモリと上記セレクタとの間のアクセスバスの本数よりも多くする。上記複数のインターフェース部の上記プロセッサは、上記2重化された共有メモリ部に2重ライトを行う。

【効果】メモリアクセスの応答時間の短縮、処理トランザクション数の増加を図れる。



(2) 000-267815 (P2000-267815A)

**【特許請求の範囲】**

【請求項1】それぞれプロセッサを有する、ホストコンピュータ又はディスク装置との複数のインターフェース部と、

それぞれ、制御情報を格納する共有メモリを有し、上記複数のインターフェース部の各インターフェース部との間がそれぞれアクセスパスにより1対1に接続される2重化された共有メモリ部と、

上記複数のインターフェース部に接続されるセレクタと、上記セレクタに接続されるキャッシュメモリ、とを有し、

上記複数のインターフェース部と上記セレクタとの間のアクセスパスの本数は、上記キャッシュメモリと上記セレクタとの間のアクセスパスの本数よりも多く、

上記複数のインターフェース部の上記プロセッサは、上記2重化された共有メモリ部に2重ライトを行うことを特徴とするディスクアレイ制御装置。

【請求項2】上記複数のインターフェース部と上記共有メモリ部とに接続されるセレクタを有し、

上記複数のインターフェース部と、上記複数のインターフェース部と上記共有メモリ部とに接続される上記セレクタとの間のアクセスパスの本数は、上記キャッシュメモリと、上記複数のインターフェース部と上記共有メモリ部とに接続される上記セレクタとの間のアクセスパスの本数よりも多いことを特徴とする請求項1に記載のディスクアレイ制御装置。

【請求項3】それぞれプロセッサを有する、ホストコンピュータ又はディスク装置との複数のインターフェース部と、

それぞれ、制御情報を格納する共有メモリを有し、上記複数のインターフェース部の各インターフェース部との間がそれぞれ2系統化されたアクセスパスにより1対1接続される2重化された共有メモリ部とを有し、

上記インターフェース部の各インターフェース部は、該インターフェース部内の上記プロセッサから上記2重化された共有メモリ部に対する2重ライトの要求を受け取った場合、対応するデータを、上記2重化された共有メモリ部の各共有メモリ部に対し交互に送信する手段を有することを特徴とするディスクアレイ制御装置。

【請求項4】上記インターフェース部は、2重ライトに関しては最初にライトする方の共有メモリの面をアドレス毎に異なる共有メモリに割り当てるなどを特徴とする請求項3に記載のディスクアレイ制御装置。

【請求項5】上記複数のインターフェース部の各インターフェース部は、上記2系統化された各アクセスパス対応に複数のキューを有し、

上記プロセッサの要求に応じて上記共有メモリにアクセスする際に、上記複数のキューにキューイングされている処理待ちの数に応じて、上記2系統化されたアクセスパスの何れか一方のアクセスパスを選択する選択手段を

有することを特徴とする請求項2乃至4の何れかに記載のディスクアレイ制御装置。

【請求項6】上記選択手段は、上記複数のキューにキューイングされている処理待ちの数が少ない方のアクセスパスを選択することを特徴とする請求項5に記載のディスクアレイ制御装置。

【請求項7】上記複数のインターフェース部の各インターフェース部は、上記キューにキューイングされている処理待ちの2重ライト、又は2重アトミックモディファイの数に応じて、次の処理要求を上記キューにキューイングするか否かを制御する制御手段を有することを特徴とする請求項5に記載のディスクアレイ制御装置。

【請求項8】上記制御手段は、上記キューに格納されている処理待ちの2重ライト、2重アトミックモディファイの数が一定数以下であれば、次の処理要求を、上記キューの内、上記2重ライト、又は2重アトミックモディファイの処理がキューイングされていないキューにキューイングすることを特徴とする請求項7に記載のディスクアレイ制御装置。

【請求項9】上記2系統化されたバスの内、1系統が故障した場合、他の1系統に処理を割り当てるなどを特徴とする請求項3乃至請求項8の何れかに記載のディスクアレイ制御装置。

**【発明の詳細な説明】****【0001】**

**【発明の属する技術分野】**本発明は、ホストコンピュータ又はディスク装置とのインターフェース部を制御する複数のCPUと、制御情報を格納する複数のメモリを含んで構成されるディスクアレイ制御装置に関する。

**【0002】**

**【従来の技術】**共有メモリ型マルチプロセッサシステムを用いるディスクアレイ制御装置として、図21に示す構成のものが知られている。図21に示す制御装置では、複数のCPU-PK(パッケージ)2101と、制御情報を格納する共有メモリを搭載する共有メモリパッケージ(SM-PK) #A 2103、及び共有メモリパッケージ(SM-PK) #B 2104とが、共有メモリバス2102を介して接続されている。各CPU-PKは、ホストコンピュータ又はディスク装置の何れかに接続される。各CPU-PKは、複数のCPUを有しており、各CPUは、共有メモリに格納されている制御情報を用いて、ホストコンピュータ又はディスク装置からのデータ転送、又はホストコンピュータ又はディスク装置へのデータ転送を制御等を行う。

【0003】また、従来のメモリ制御装置では、メモリアクセスの応答時間の短縮、処理トランザクション数の増大を図るため、複数のCPUとメモリとの間を複数のバスで接続したものがある。また、処理トランザクション数を増加させるには、処理トランザクションをそれら複数のバスに最適に振り分ける必要がある。トランザク

:(3) 000-267815 (P2000-267815A)

ションを振り分ける技術に関しては、特開平9-146863号公報に、各バスの使用、未使用的フラグを見てトランザクションを振り分ける旨が開示されている。

【0004】また、従来のメモリ制御装置では、信頼性を向上させるため、バスを2系統化、メモリを2重化したものがある。このようなメモリ制御装置は、例えば、特開平9-325916号公報に、メモリアクセスに関して2つのバスを使うことにより、処理トランザクション数を増大させ、同時にデータの信頼性を向上するために重要なデータは2個のメモリに2重化ライトしている。この様な方式では同期複数メモリに同一データを書くため、同時に複数の2重ライトが発生するとデータの不一致またはデットロックを起こす可能性がある。ここでは2重ライトについて、メモアドレス毎に使用するバスを固定することによりアクセス経路を单一化し、データの不一致またはデットロックの問題を避けている。

【0005】しかし、このような方式では、多数のアクセスが発生した場合、バスでの処理がシリアルになり応答時間、処理トランザクション数に関して性能ボトルネックとなる。またバスを2重化したとはいえ、バスの故障が発生した場合、接続しているCPUはすべて影響を受けることになる。

【0006】性能、信頼性の向上を図ったものとしてと、CPUとメモリの間を一对一に接続するスター接続方式がある。特開昭58-16362号公報には、各CPUから2重化された各メモリに対し、1本づつバスが接続されている。この方式は、上述の方式に比較して同時にアクセスできるバス数が増え、処理トランザクション数もさらに増大する。また、各CPU毎にバスが接続されているので、本方式は、上述の方式に比較し、1つのバスの故障により影響を受けるCPUの範囲は狭く、信頼性も高い。さらに、この技術では、2重ライト処理時のデータの不一致またはデットロックを避けるために、共有メモリ間インターフェイスをもうけている。

【0007】

【発明が解決しようとする課題】ディスクアレイ制御装置において、共有メモリではキャッシュメモリほど大量のデータの転送は必要としないが、トランザクション数を増加させ、一回の転送の応答時間の短縮が必要である。したがって、特開昭58-16362号公報に記載されている技術のように、各CPUと共有メモリとの間を1対1接続するのが適している。

【0008】一方、各CPUとキャッシュメモリとの間では、大量のデータを高速に転送する必要があるので、これらの間のアクセスバス数を増やす必要がある。そのためには、特開昭58-16362号公報に記載されている技術のように、これらの間を1対1接続するのが適している。

【0009】しかし、キャッシュメモリを構成するバッ

ケージに実装できるコネクタ数、又は、各メモリを構成するLSIに実装できるピン数の数には物理的に限りがあるので、各CPUと共有メモリとの間のアクセスバス数を増やすことには限界がある。

【0010】また、上述のように、処理トランザクション数を増加し、信頼性を向上するには、バス接続方式よりスター接続方式の方が優れているが、特開昭58-16362号公報に開示された技術では、2重ライト処理用に共有メモリ間にインターフェイスを設けているため、このインターフェイスに障害が発生すると共有メモリ全体が使用できなくなるという問題がある。また、2重ライト時には両方の共有メモリをすべてロックして他のアクセスを拒絶するため、処理トランザクション数が減少するという問題もある。さらに、バスの本数を単純に増やしても、先ほどのバス方式の様に2重ライトをシリアルに処理する必要があるので、トランザクション数はそれほど増加しない。また各CPUと共有メモリ間のバスは1本なので、そのバスに障害が発生した場合、データにアクセスできなくなる可能性がある。

【0011】そこで、本発明の目的は、複数のCPUと2重化された各メモリとの間を、メモリアクセスの応答時間の短縮と、処理トランザクション数を増加させることにある。

【0012】また、2重ライト時にデータの不一致またはデットロックを起こさないバスの選択制御手段を提供することにある。

【0013】

【課題を解決するための手段】上記問題を解決するため、本発明はCPUと各共有メモリバスを接続し、制御を行う共有メモリバスインターフェイス部に各バスの使用状況を示すバスロックテーブルを設け、共有メモリと各共有メモリバスを接続し制御する共有メモリ制御回路にメモリのロック情報を示すメモリロックテーブルを設ける。2重ライト以外の処理（1重リード、1重ライト、アトミックモディファイ）の場合は各バスの中で処理待ちトランザクション数を示す、バスキー長の短いバスを選択する。2重ライト、2重アトミックモディファイの場合、バスを選択する時に上記バスロックテーブルを参照し、既に処理中の2重ライト数が一定数以下なら2重処理を開始し一定数以上なら待つ。さらに2重ライトするバスを選択する際に、すでに処理中の2重ライト処理とは異なるバスで処理を行う様にバスを選択する。

【0014】

【発明の実施の形態】以下、本発明の実施例について説明する。

【0015】まず、最初に、本発明のディスクアレイ制御装置の全体構成例を図2を用いて説明する。ディスクアレイ制御装置610は、上位装置500と、磁気ディスク570、580等に接続されている。500は汎用

:(4) 000-267815 (P2000-267815A)

コンピュータ、ワークステーション、パーソナルコンピュータに代表される上位装置（ホスト）であり、二次記憶装置に対してリード、ライト等の命令を発行し、結果を受けとる。CHA-PK510、520はチャネルアダプタパッケージであり、上位装置500からの命令を受け取り、キャッシュ550、共有メモリ560、DKA-PK530、540をアクセスする。DKA-PK530、540はディスクアダプタパッケージであり、共有メモリ560またはCHA-PK510、520からの命令により、磁気ディスク570、580、590、600をアクセスする。ここでCHA-PK510、520、DKA-PK530、540は図1のCPU-PK10等と同じ構成になっている。511、512、513、514、521、522、523、524、533、534、543、544はCPUであり各パッケージを制御する。DDTA515、525、533、543は、上位装置500または磁気ディスク570、580、590、600とキャッシュ550との間のデータの送受信を行う。MPA516、526、534、544は、各CPUと共有メモリ560との間の共有データのやり取りを制御する。550はキャッシュであり、上位装置500または磁気ディスク570、580、590、600のデータを一時的に格納する。Select or 551、552、553、554は、DDTA515、525、533、543とCache555、556、557、558との間の接続を制御する。555、556、557、558はCacheであり、各Select orから来るデータを格納する制御回路とメモリである。560は共有メモリであり上位装置500との応答、キャッシュ550、磁気ディスク570、580、590、600のデータのアクセスに関する制御データを格納する。561、562はSMA-A、Bであり、各MPAから来る制御データを格納する制御回路とメモリである。ここでSMA-A561、SMA-B562は図1のSMA-PK#A120、SMA-PK#B140と同じ構成を取っている。570、580、590、600は磁気ディスクであり、データを格納し必要とあればキャッシュ550にデータをロードする。

【0016】DDTA515、525等とCache555、556等との間では、大量のデータを高速に転送する必要があるので、DDTA515、525等とCache555、556等との間のアクセスバス数を増やす必要がある。そのためには、DDTA515、525等とCache555、556等との間を1対1接続するのが適している。しかし、Cache555、556等を構成するパッケージに実装できるコネクタ数、又は、キャッシュ113を構成するLSIに実装できるビン数の数には物理的に限りがあるので、DDTA515、525等とCache555、556等との間のアクセスバス数を増やすことには限界がある。そこで、D

DTA515、525等とCache555、556等との間にセレクタ551、552等を設け、DDTA515、525等とセレクタ551、552等との間を1対1接続することにより、DDTA515、525等とセレクタ551、552等との間のアクセスバス数を増やす。一方、セレクタ551、552等で、複数のDDTA515、525等からのアクセス要求を所定数に絞ることにより、Cache555、556等とセレクタ551、552等との間のアクセスバス数を、DDTA515、552等とCache555、556等との間のアクセスバス数よりも減らし、上述のコネクタ数、又はビン数の問題を解決している。

【0017】一方、共有メモリではキャッシュメモリほど大量のデータの転送は必要としないが、トランザクション数を増加させ、一回の転送の応答時間の短縮が必要である。そこで、SMA-A561、SMA-B562と各CPU-PKとの間は、セレクタ部での遅延を避けるため、セレクタを介さずに接続している。ただし、MPAとSMA-A、SMA-Bとの間にも、セレクタを設ける構成してもよい。

【0018】なお、以下の説明で利用するディスクアレイ制御装置の構成図では、キャッシュメモリを省略している。

【0019】[実施例1] 図1は、本発明のディスクアレイ制御装置の一実施例を示している。

【0020】10、40、50はCPUパッケージ(CPU-PK#1、#2、#3)であり、複数のCPU、メモリ等からなる。15、18はCPUであり各メモリのリード、ライト、演算、命令実行を行う。20、23はローカルメモリであり各CPUに必要な情報、プログラムを格納する。25、28はローカルバス回路でありCPU15、ローカルメモリ20、共有メモリインターフェイス30を接続し、互いのデータのやり取りを管理する。30は共有メモリインターフェイス(MPA)であり、CPU15、18等と共有メモリバス(60~115)との間を接続しデータのやり取りを制御する。共有メモリインターフェイス30は内部にバスロックテーブルを持ち、バスの選択、排他処理を行う際に用いる。60、65、70、75、80、85、90、95、100、105、110、115は共有メモリバスであり、CPU15、18と共有メモリ130、150との間で送信されるデータは、この共有メモリバスを用いて送信される。120はSMA-PK#A、140はSMA-PK#Bであり、各MPAから来る制御データを格納する制御回路とメモリである。SMA-PK#A120とSMA-PK#B140はメモリの一部が互いに対となっており、2重化されている。125は共有メモリ制御回路A(SMA-A)、145は共有メモリ制御回路B(SMA-B)であり各CPUパッケージからくる共有メモリアクセスを処理する。126、146はメモリロ

!(5) 000-267815 (P2000-267815A)

ックテーブルであり、共有メモリのロック情報を管理する。130、150は共有メモリであり、各CPU間で共有する制御情報等を格納する。

【0021】図3は、図1の構成を更に詳細に説明したものである。

【0022】CPU-PK#1(10)内の共有メモリバスインターフェイス(MPA)内には、バスロックテーブル31以外に、各バス毎に処理待ちトランザクションを登録するキューがある。32はA面バス0キュー、33はA面バス1キュー、34はB面バス0キュー、35はB面バス1キューである。MPA30はこのキュー並んでいる順番通りにトランザクションを各バスに通す。またこのキューで処理を待っているトランザクションの数をバスキュー長としてカウントし、バスロックテーブル31に記録する。また現在処理中のアクセスの種類や、キュー内にある2重アクセスの数等もバスロックテーブル31に記録する。CPU-PK10からSMA-PK125、145へは各2本のバスがあり、よって最大で同時に4回のアクセスが処理できることになる。各バスからのアクセス要求を受けた共有メモリ制御回路-A125、共有メモリ制御回路-A145はアクセス要求の優先順位を決めて、共有メモリ130、150のアクセスを行う。その際メモリロックテーブルにアクセス中のアドレスを記憶し、他のアクセスがそのアドレスを後から要求しても排他する。また2重ライトの場合、A面の共有メモリ130、B面の共有メモリ150の内、先にライトする方をマスタとし後でライトする方をスレーブとする。この順番はユーザが設定したデータ長によってマスタ、スレーブが入れ替わる。図3の例ではA面の0x00..000から0xA0..000までがマスタ面であるとすると、次の0xB0..000まではスレーブ面となる。その時B面は0x00..000から0xA0..000までがスレーブ面であり、次の0xB0..000まではマスタ面となる。この方式によって従来ではシリアルに処理していた2重ライトを、条件によっては並列に処理することが可能となる。図3の例では2重ライト、リード領域についてのみ示しているが、1重ライト、リード領域の場合A面とB面では異なるアドレスが割り当てられるので、上記に示したようなマスタ、スレーブの設定はない。

【0023】図4を用いて、本発明の実施例1を適用した場合の利点を示す。2重ライトが効率よく処理できる例では、2個の2重ライトに対し、互い異なる共有メモリがマスタに割り当てられた場合に、マスタ面のアクセス、スレーブ面のアクセスに対し、4本のバスを使い分けることによってほぼ平行してアクセスが可能になる。よって処理トランザクション数を増加させることが可能になる。ただし、2重ライトでデッドロックが生じる例にあるようにバスの選択を間違えると、お互いに相手が使いたいバスを占有し続けデッドロックが生じ、処理は前に進まなくなる。よってバスロックテーブル31やメ

モリロックテーブル126、146を用いてデッドロックの無いバスの選択を行う必要がある。

【0024】図5は、バスロックテーブル31を示す。400は共有メモリ面であり、バスの接続している共有メモリ面を示している。405は使用バス番号であり、システム全体で何番目のバスなのかを示す。410はバスキー長でバスのキーで処理を順番待ちしているトランザクション数を示している。415はキュー内の2重アクセスであり、処理中または待ち行列に並んでいるトランザクションの中に2重ライトが含まれているかどうかを示している。420は処理中のアクセス種類であり、処理中のアクセスが1重リード、1重ライト、2重ライト、アトミックモディファイ(リード、ライト)、2重アトミックモディファイ(リード、ライト)のどれであるかを示す。425は処理中のロックの有無であり、アトミックモディファイの様に処理中のリードの終了後もバスをロックし続けるか否かを示す。430は処理中のマスタ、スレーブの有無を示し、処理中のアクセスがマスタ側なのか、スレーブ側なのかを示している。

【0025】図6は、メモリロックテーブル126、146を示す。450は使用バス番号であり、システム全体で何番目のバスからのアクセスなのかを示す。455は排他開始アドレスであり、アクセスしている共有メモリの先頭アドレスを示す。460は排他終了アドレスであり、アクセスしている共有メモリの終端アドレスを示す。排他開始アドレス455と排他終了アドレス460の範囲のデータは排他となり、他のアクセスはできないことになる。また排他するデータ長はユーザが前もって設定する。465はロックアクセスの有無であり、マスタ、アトミックモディファイの場合は共有メモリ130、150のアクセス終了後も排他を解除しないことを示す。アクセスが完全に終了後、共有メモリバスインターフェイス30からの命令によって初めてこのロックは解除され、ロックアクセスの有無の欄は空欄になる。

【0026】図7に本発明の実施例1における1重ライト、1重リードの動作を示す。以下特に断らない限り図2の構成を例に説明する。(1) 1重ライトの動作ではアドレス、コマンド、データはCPU15、18からローカルバス回路25、28を通してMPA30に送信され、MPA30から共有メモリバスの0番目60、70または1番目65、75のバスを選択してSMA-PK#A120、SMA-PK#B140の共有メモリ制御回路125、145へ送られる。ライト処理が終了すると終了情報を乗せたステータスがSMAから同じ共有メモリバスを通してMPA30に送信され、MPA30からローカルバス回路25、28を通して次の送信が可能であることを示すレディがCPU15、18に戻される。(2) のリードの動作ではアドレス、コマンドはCPU15、18からローカルバス回路25、28を通してMPA30に送信され、MPA30から共有メモリバ

(6) 000-267815 (P2000-267815A)

スの0番目60、70または1番目65、75のバスを選択してSMAの共有メモリ制御回路125、145へ送られる。リード処理が終了するとデータ、ステータスがSMAの共有メモリ制御回路125、145から同じ共有メモリバスを通してMPA30に送信され、MPA30からデータとレディがCPU15、18に戻される。1重ライト、1重リードの動作中MPA30はSMA-PK#A120、SMA-PK#B140からステータスが返ってくるまで共有メモリバスを切り離さず保持している。

【0027】図8に2重ライトの動作を示す。これ以降、特に断りがなければ、A面共有メモリ130をマスタとして話を進めるが、これはB面共有メモリ150がマスタになった場合でも同様である。(3) 2重ライトの動作ではアドレス、コマンド、データはCPUからローカルバスを通してMPA30に送信される。MPA30からはマスタになる面(図9の例ではA面)の0番目60または1番目65のバスを選択してSMA-PK#A120の共有メモリ制御回路-A125に送信される。ライト処理が終了すると終了情報を乗せたステータスがSMA-PK#A120の共有メモリ制御回路-A125から同じ共有メモリバスを通してMPA30に送信される。この後もマスタ側のバスは共有メモリ制御回路-A125が保持し、他のアクセスはこのバスを使うことができない。ステータスが正常ならば、MPA30は次にスレーブになる面(図9の例ではB面)の0番目70または1番目75のバスを選択してSMA-PK#B140の共有メモリ制御回路-B145にアドレス、コマンド、データを送信する。ライト処理が終了すると終了情報を乗せたステータスがSMA-PK#B140から同じ共有メモリバスを通してMPA30に送信される。スレーブ側は転送終了後バスをすぐに開放する。ステータスが正常ならば、MPA30が保持していたマスタ側のバスを開放するためのアンロック命令が共有メモリ制御回路-A125に送信され、マスタ側のバスが開放される。次にローカルバスを通して次の送信が可能であることを示すレディがCPU15、18に戻される。

【0028】図9に、1重アトミックモディファイ、2重アトミックモディファイの動作を示す。アトミックモディファイは、共有メモリのリードの後に他のアクセスを介入させることなくライトを行う処理で、マルチプロセッサの同期化等に必要な機能である。(4) 1重アトミックモディファイの動作では、アドレス、コマンドはCPU15、18からローカルバスを通してMPA30に送信され、MPA30から共有メモリバスの0番目60または1番目65のバスを選択してSMA-PK#A120(この例では共有メモリ制御回路-A125)へ送られる。リード処理が終了するとデータ、ステータスがSMA-PK#A120から同じ共有メモリバスを通してMPA30に送信され、MPA30からデータとス

テータがCPU15、18に戻される。この後もリードで用いたバスは共有メモリ制御回路-A125が保持し、他のアクセスはこのバスを使うことができない。ステータスが正常でCPU15、18がモディファイ処理を行った後、MPA30は次にSMA-PK#A120の共有メモリ制御回路-A125にアドレス、コマンド、データを送信する。ライト処理が終了すると終了情報を乗せたステータスがSMAの共有メモリ制御回路-A125から同じ共有メモリバスを通してMPA30に送信される。また同時に今までロックしていたバスも開放する。次にMPA30から次の送信が可能であることを示すレディがCPU15、18に戻される。(5) 2重アトミックモディファイ動作では、前半のリードは1重アトミックモディファイと同じである。ステータスが正常でCPU15、18がモディファイ処理を行った後、MPA30はすでにバスを確保しているマスタ側共有メモリ130とまだ確保していないスレーブ側共有メモリ150に対して、2重ライトの処理を行う。MPA30からアドレス、コマンド、データはマスタ側バスを使ってSMA-PK#A120の共有メモリ制御回路-A125に送信される。ライト処理が終了すると終了情報を乗せたステータスがSMAの共有メモリ制御回路から同じ共有メモリバスを通してMPA30に送信される。スレーブ側では、0番目70または1番目75のバスを選択してSMA-PK#Bの共有メモリ制御回路B-145にアドレス、コマンド、データを送信する。ライト処理が終了すると終了情報を乗せたステータスがSMA-PK#Bから同じ共有メモリバスを通してMPA30に送信される。スレーブ側は転送終了後バスをすぐに開放する。ステータスが正常ならば、MPA30が保持していたマスタ側のバスを開放するためのアンロック命令が共有メモリ制御回路-A125に送信され、マスタ側のバスが開放される。次にローカルバスを通して次の送信が可能であることを示すレディがCPU15、18に戻される。図7、8、9で示した各アクセスを実現するために共有メモリバスインターフェイス(MPA)30と共有メモリ制御回路A(SMA-A)125、共有メモリ制御回路B(SMA-B)145で以下に述べる様な、処理フローが必要である。

【0029】図10に、バスキュー振り分け処理のフローを示す。これは共有メモリバスインターフェイス(MPA)30においてCPU15、18等からトランザクションを受け取り、MPA30の各バスキューに投入するまでのフローである。700からバスキュー振り分け処理が開始される。ステップ701では複数のCPU15、18等から来る処理をラウンドロビン等を使って優先順位をつけて、一番優先度の高い処理を選択する。ステップ702ではその処理がSMA-PK#A120(A面)処理から開始するかを判定する。もしSMA-PK#B120(B面)から処理が開始されるのならば

(7) 000-267815 (P2000-267815A)

ステップ703のアクセス処理に進む。もしA面から処理が開始されるのならば、ステップ710のアクセス処理へ進む。ここでステップ703とステップ710はA面、B面の違いを除いて、処理内容はまったく同じである。ステップ711では処理が2重ライトまたは2重アトミックモディファイであるかを判定する。もし、2重ライトまたは2重アトミックモディファイならばステップ730の2重ライトまたは2重アトミックモディファイ処理に進む。ステップ730については後で説明する。もし2重ライトまたは2重アトミックモディファイでないならばステップ720のキュー選択処理に進む。ステップ720内のステップ721ではMPA30内のバスロックテーブル31のバスキュー長410をA面に関して見る。ステップ722ではバス0(60)とバス1(65)のバスキュー長410を比較する。もしバス0(60)の方が長ければ、ステップ723に進み、バスキュー1に処理を入れる。次にステップ724に進み、バスロックテーブル31内のバス1キュー長を1加算し、処理の登録を行う。もしバス1(65)の方が長ければ、ステップ727に進み、バスキュー0に処理を入れる。次にステップ727に進み、バスロックテーブル31内のバス1キュー長を1加算し、処理の登録を行う。もしどちらの長さも同じならば、ステップ725に進み、前回CPUが使わなかったバスキューに処理を入れる。次にステップ728へ進み、バスロックテーブル31内の該当するバスキュー長を1加算し、処理の登録を行う。ステップ723、726、728の後はステップ701に戻る。ステップ703のB面のアクセス処理が終了した後も同様にステップ701へ戻る。以上のキュー選択処理710により、CPU15、18の出す処理は2本のバスのバスキュー長の短い方へ入る制御が行われるので、バス効率が向上する。

【0030】図11に、2重ライト、2重アトミックモディファイ処理のフローを示す。ステップ731ではアクセスがマスタ側の処理なのかを判定する。もしスレーブ側の処理ならば、ステップ734へ進む。もしマスタ側の処理ならばステップ732に進みバスロックテーブル31のキュー内の2重アクセス415、処理中のマスター/スレーブの有無430を調べ、マスタ側の処理および2重アトミックモディファイの個数を数える。ステップ733ではマスタ数が2以下であるかを判定する。2以上であればすでに2個以上マスタ側の処理が入っており、これ以上2重ライト、2重アトミックモディファイの処理是不可能である。よってステップ732に戻り、マスタ数が1以下になるまで待つ。もしマスタ数が1以下になったら、ステップ734に進む。ステップ734ではバスロックテーブル31より、アクセスする共有メモリ面（該当面）のバスが2本とも正常で、かつどちらかにマスタ側の処理があるかを調べる。もしどちらかにマスタ側の処理があるならば、ステップ735に進み、

マスタがないバスに処理を入れる。次にステップ736に進み、バスロックテーブル31内の該当バスのバスキュー長410を1加算してステップ737へ進む。もしマスタがどちらにも無ければ、ステップ720に進みキュー選択処理を行い、入れるバスを確定しステップ737に進む。ステップ737ではバスロックテーブルのキュー内2重アクセス415を登録する。以上の2重ライト、2重アトミックモディファイ処理730によってバスを確定することにより、デッドロックなしに2重ライト、2重アトミックモディファイのアクセスを最適にキューに振り分けることが可能になる。

【0031】図12に、各バスキューの処理のフローを示す。これは各キューに入った処理を各バスを通してSMA-PK#A120、SMA-PK#B140に送信し、その返答があれば、それをCPU15、18等に返す処理である。ステップ750から各バスキューの処理が開始される。ステップ751ではバスキューより実行可能な処理を取り出す。ステップ752では取り出した処理についてバス毎にSMAアクセス処理を行う。SMAアクセス処理はSMAに対して図7、8、9のようなプロトコルでコマンド等を発行し、データ、ステータス等を受け取ることである。ステップ753では返ってきたステータス等より処理が正常終了したかを判定する。もし正常終了ならば、ステップ760のコマンド終了処理へ進む。もし異常終了ならば、ステップ754の障害処理を行い、ステップ760へ進む。障害処理の詳細については本発明とは直接関係ないのでこれ以上の説明は省略する。ただしバス障害処理については後で説明する。ステップ760のコマンド終了処理については後で説明する。コマンド終了処理760が終了した後ステップ751に戻る。

【0032】図13にコマンド終了処理を示す。SMAアクセス処理が終了した後、バスロックテーブル31の処理中アクセス種類420、処理中のロックの有無425、処理中のマスター/スレーブの有無430を調べその組み合わせによってコマンド終了処理760にある次に行う処理を選択する。次に行う処理は図7、8、9のプロトコルを実現するために必要な処理である。

【0033】以上図10、11、12で説明した処理をMPA30の中で行う。MPA30から発行されたコマンド等はSMA-PK#A120、SMA-PK#B140に送信されて、処理が行われる。

【0034】図14に、SMA内各バス処理のフローを示す。これはSMA-PK#A120、SMA-PK#B140に到着した各バスからの処理に優先順位をつけメモリロック等を行い、共有メモリのアクセス処理を実行、その後終了処理を行いMPA30に終了報告を行までのフローである。ステップ800からSMA内各バス処理が開始される。ステップ801では各バスに到着した処理のデコード、エラーのチェックを行う。ステッ

!(8) 000-267815 (P2000-267815A)

ア802では到着した処理が2重ライト、アトミックモディファイ等の最後発生するアンロック以外かどうかの判定を行う。もしアンロック以外であればステップ803に進む。アンロックであれば図15の4へ進む。ステップ803では各バスに到着した処理の実行順位をラウンドロビン等で決定する。ステップ804では各バスに有る処理がランドロビンの結果実行可能かどうかを判定する。実行可能でない場合再度ステップ803に戻る。実行可能ならば、ステップ805に進み、メモリロックテーブル126、146の排他開始アドレス455、排他終了アドレス460を参照してすでに排他処理中のアドレスと重なるかどうかを検索する。アドレスが重なっていればその領域はロックされていることになる。ステップ806では検索結果よりアクセス先が未ロック領域であるかどうかを判定する。もしロック領域ならば、ロックが開放されるまで待つ。もし未ロック領域ならば、ステップ807に進み、メモリロックテーブル126、146に排他開始アドレス455、排他終了アドレス460で示されるアクセス範囲、ロックアクセスの有無等を登録する。その後ステップ810では共有メモリのアクセス処理を行い共有メモリ130、150のリード、ライト等が処理される。共有メモリ130、150の処理が終わった後のフローについては図14、15の2以降に示す。

【0035】図15に、SMA内各バス処理（続き）のフローを示す。ステップ820では共有メモリ130、150からデータ、ステータスを受け取る。ステップ821ではデータ、ステータスより、処理が正常終了したかどうかを判定する。もし正常終了ならば、ステップ823へ進む。もし異常終了ならば、ステップ822の障害処理を行い、ステップ823へ進む。障害処理の詳細については本発明とは直接関係ないのでこれ以上の説明は省略する。ステップ823では終了した処理がロックアクセス処理であるのかを判定する。もしロックアクセスならば、そのままメモリ、バスをロックしたままで、ステップ825へ進む。もしロックアクセスでないならば、ステップ824に進み、メモリロックテーブル126、146に登録した使用バス番号450、排他開始アドレス455、排他終了アドレス460、ロックアクセスの有無465を削除する。その後ステップ825に進む。ステップ825では処理の正常終了、異常終了等のステータスを生成してMPA30に送信する。

【0036】図16に、共有メモリのアクセス処理のフローを示す。ステップ810より共有メモリのアクセス処理が開始される。ステップ811ではSMA内各バスよりメモリアクセスを受け取る。ステップ812では先行している処理があるかどうかを判定する。もしなければ、ステップ818に進む。もしあればステップ813に進み、アクセスが終了している処理があるかどうかを判定する。もしなければ、ステップ814に進みメモリ

多重アクセスが可能かどうか判定する。もし可能でなければそのままステップ813にもどる。もし可能ならばステップ818に進む。アクセスが終了している処理があれば、ステップ815に進み、その処理が正常終了したかどうかを判定する。もし正常終了ならば、ステップ7817へ進む。もし異常終了ならば、ステップ816の障害処理を行い、ステップ817へ進む。障害処理の詳細については本発明とは直接関係ないのでこれ以上の説明は省略する。ステップ817では処理の正常終了、異常終了等のステータスを生成してMPA30に送信する。ステップ818では共有メモリへのリード、ライトを実際に実行する。その後ステップ811に戻る。

【0037】以上、図10から図16に示した、MPA30とSMA-PK#A130、SMA-PK#B150の処理フローを実現することにより、2重ライト時にデータの不一致またはデットロック無しで効率の良いバス選択ができる。また図17で示すようなバス障害処理を行することで、バスの2重化が実現でき、信頼性が向上する。

【0038】図17に本発明の実施例1におけるバス障害処理のフローを示す。ステップ850からバス障害処理が開始される。ステップ851ではエラー情報の収集をステータス等から集める。ステップ852ではバス障害であるかどうかを判定する。もしバス障害でなければ他の障害処理を行う。もしバス障害であれば、ステップ854に進み障害バスの特定を行う。ステップ855では障害バスを利用する各CPU15、18、SMA-PK#A130、SMA-PK#B150に対してバス閉塞を報告する。ステップ856ではバスロックテーブル3内の使用バス番号を変更する。ステップ857では障害バスを論理的、物理的に閉塞を行う。ステップ858でバス障害処理は終了する。

【0039】以上図1から図17に示した実施例1によれば、2重化したバスをスター型に接続した2重化共有メモリ制御装置において、共有メモリバスインターフェス側にバスロックテーブルを、共有メモリ制御回路側にメモリロックテーブルを持ち、これを参照しながらバスを選択することにより、1重系アクセスだけでなく、2重ライト、2重アトミックモディファイのバス選択をデットロック無しに効率よくできるので、処理待ち時間が短縮し、処理トランザクション数の増加を図ることが可能になる。

【0040】【実施例2】次に、本発明のディスクアレイ制御装置の他の実施例について説明する。以下、特に説明のない部分は実施例1と同じとする。

【0041】図18に、本実施例の基本構成を示す。ここで実施例1と異なる所は共有メモリ130、150におけるマスタスレーブの配置の方法である。実施例2では共有メモリ120、140のどちらか片面をすべてマスタとしている。これにより2重ライト、2重アトミック

(9) 000-267815 (P2000-267815A)

クモディファイのバス選択は必ずマスタになった共有メモリの面から始まるので、デッドロックの可能性が無くなり、制御も楽になる。ただし、これらのアクセスがすべてシリアルな処理になるので、トランザクション数の増加は実施例1に比較して少なくなる。

【0042】図19に本発明の実施例2における2重ライト、2重アトミックモディファイ処理のフローを示す。この部分は図11の2重ライト、2重アトミックモディファイ処理730に対応する部分である。ステップ921ではアクセスマスタ側の処理かどうかを判定する。もしスレーブ側の処理であれば、ステップ924に進む。もしマスタ側であれば、ステップ922に進みバスロックテーブル31より全キュー内のマスタ処理数を数える。ステップ923ではマスタ処理数が2以下かどうか判定する。2以上ならば、これ以上2重ライト、2重アトミックモディファイ処理は入れないので、ステップ922に戻り、待つ。もしマスタ数が1以下であれば、処理が可能なのでステップ721のキュー選択処理へ進む。ステップ924ではステップ737ではバスロックテーブル31のキュー内2重アクセス415を登録する。

【0043】以上の様な場合、2重ライトの処理時間が余計にかかる欠点があるが、もし2重ライトの中でマスタ、スレーブの順番が入れ替わっても問題ないライトがある場合、たとえば、CPU15、18のマイクロプログラムの中で2重ライトの順番が制御できる場合は以下に示すようなライトの順番を無視する2重ライトを用いることで2重ライトの処理時間を削減することが可能になる。

【0044】図20に本発明の実施例2におけるライトの順番無視する2重ライトを示す。ここでは、MPA30から送信するマスタ面へのライトとスレーブ面へのライトは同時に発生し、バスのロックも行わないで、応答時間は短縮できる。

【0045】以上図18から図20に示した実施例2によれば、2重化したバスをスター型に接続した2重化共有メモリ制御装置において、共有メモリバスインターフェス側にバスロックテーブルを、共有メモリ制御回路側にメモリロックテーブルを持ち、これを参照しながらバスを選択することにより、1重リード、1重ライト、1重リードモディファイライトの処理待ち時間が短縮し、処理トランザクション数の増加を図ることが可能になる。

【0046】本実施例では、ディスクアレイ制御装置について説明したが、上述の2重化共有メモリ装置は、高信頼性と高トランザクション性能が必要なシステム、具体的にはフォールトレントコンピュータに適用しても良い。

【0047】

【発明の効果】2重ライト時に各バスの使用状況を示す

バスロックテーブルを参照することにより、2重ライト以外の処理（1重リード、1重ライト、アトミックモディファイ）の場合は各バスの中で処理待ちトランザクション数の少ないバスを選択し、2重ライト、2重アトミックモディファイの場合、バスを選択する時に上記バスロックテーブルを参照しすでに処理中の2重ライト数が一定数以下なら2重処理を開始し、さらに2重ライトするバスを選択する際に、すでに処理中の2重ライト処理とは異なるバスで処理を行う様にバスを選択するので、データの不一致またはデットロック無しで効率の良いバス選択ができ、処理待ち時間が短縮し、処理トランザクション数が増加する。また共有メモリバスは2系統化されており、1本障害時にも共有メモリアクセスが可能になり、信頼性が向上する。

【図面の簡単な説明】

【図1】本発明のディスクアレイ制御装置の全体構成を示す。

【図2】本発明のディスクアレイ制御装置の全体構成を示す図である。

【図3】本発明のディスクアレイ制御装置の基本構成を示す図である。

【図4】本発明のディスクアレイ制御装置の利点を示す図である。

【図5】本発明のディスクアレイ制御装置におけるバスロックテーブルを示す図である。

【図6】本発明のディスクアレイ制御装置におけるメモリロックテーブルを示す図である。

【図7】本発明のディスクアレイ制御装置における、1重リード、1重ライトの動作を示す図である。

【図8】本発明のディスクアレイ制御装置における2重ライトの動作を示す図である。

【図9】本発明のディスクアレイ制御装置における1重アトミックモディファイ、2重アトミックモディファイの動作を示す図である。

【図10】本発明のディスクアレイ制御装置におけるバスキー振り分け処理のフローを示す図である。

【図11】本発明のディスクアレイ制御装置における2重ライト、2重アトミックモディファイ処理のフローを示す図である。

【図12】本発明のディスクアレイ制御装置における各バスキーの処理のフローを示す図である。

【図13】本発明のディスクアレイ制御装置におけるコマンド終了処理を示す図である。

【図14】本発明のディスクアレイ制御装置におけるSMA内各バス処理のフローを示す図である。

【図15】本発明のディスクアレイ制御装置におけるSMA内各バス処理（続き）のフローを示す図である。

【図16】本発明のディスクアレイ制御装置における共有メモリのアクセス処理のフローを示す図である。

【図17】本発明のディスクアレイ制御装置におけるバ

(10) 100-267815 (P2000-267815A)

ス障害処理のフローを示す図である。

【図18】本発明のディスクアレイ制御装置の基本構成を示す図である。

【図19】本発明のディスクアレイ制御装置における2重ライト、2重アトミックモディファイ処理のフローを示す図である。

【図20】本発明のディスクアレイ制御装置におけるライトの順番無視する2重ライトを示す図である。

【図21】従来のディスクアレイ制御装置を示す図である。

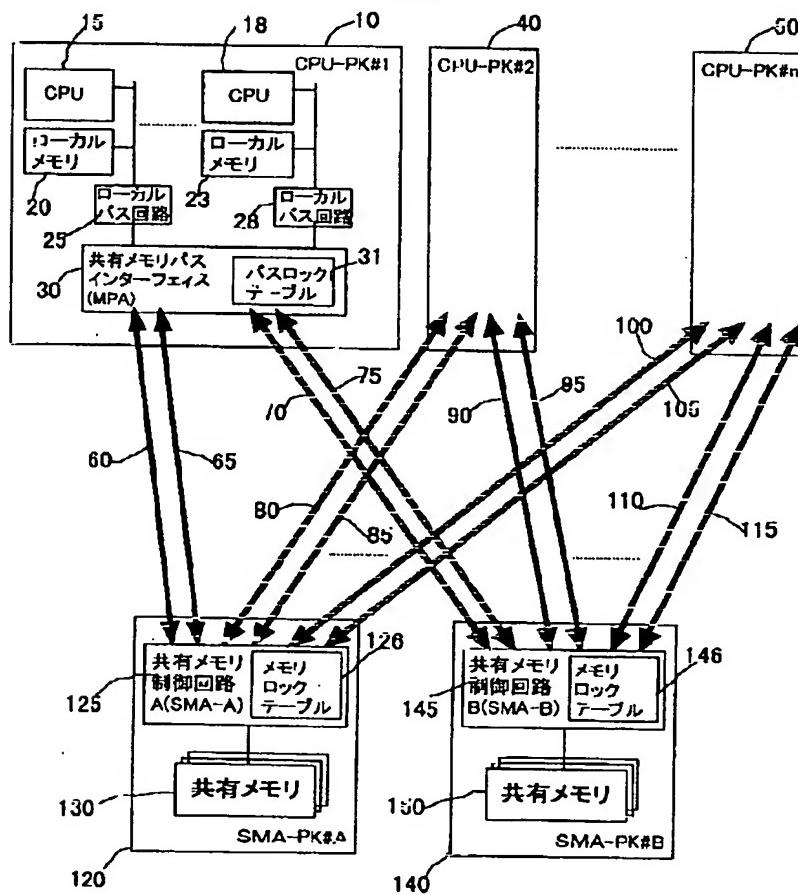
【符号の説明】

10、40、50 CPUパッケージ(CPU-PK)、

- 15、18 CPU、
- 20、23 ローカルメモリ、
- 25、28 ローカルバス回路、
- 30 共有メモリバスインターフェイス(MPA)、
- 31 パスロックテーブル、
- 60、65、70、75、80、85、90、95、100、105、110、115 共有メモリバス、
- 120 SMAパッケージA(SMA-PK#A)、
- 140 SMAパッケージB(SMA-PK#B)、
- 125 共有メモリ制御回路A(SMA-A)、
- 140 共有メモリ制御回路B(SMA-B)、
- 130、150 共有メモリ。

【図1】

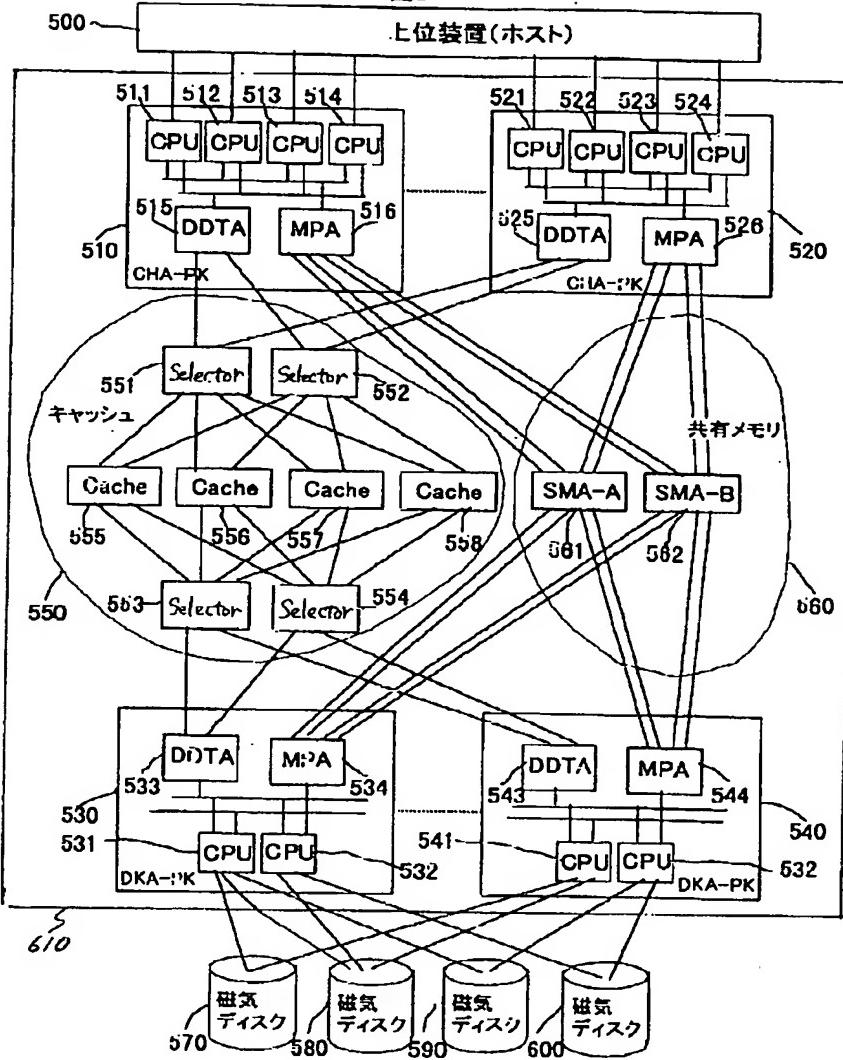
図1



(1) 00-267815 (P 2000-267815A)

【図2】

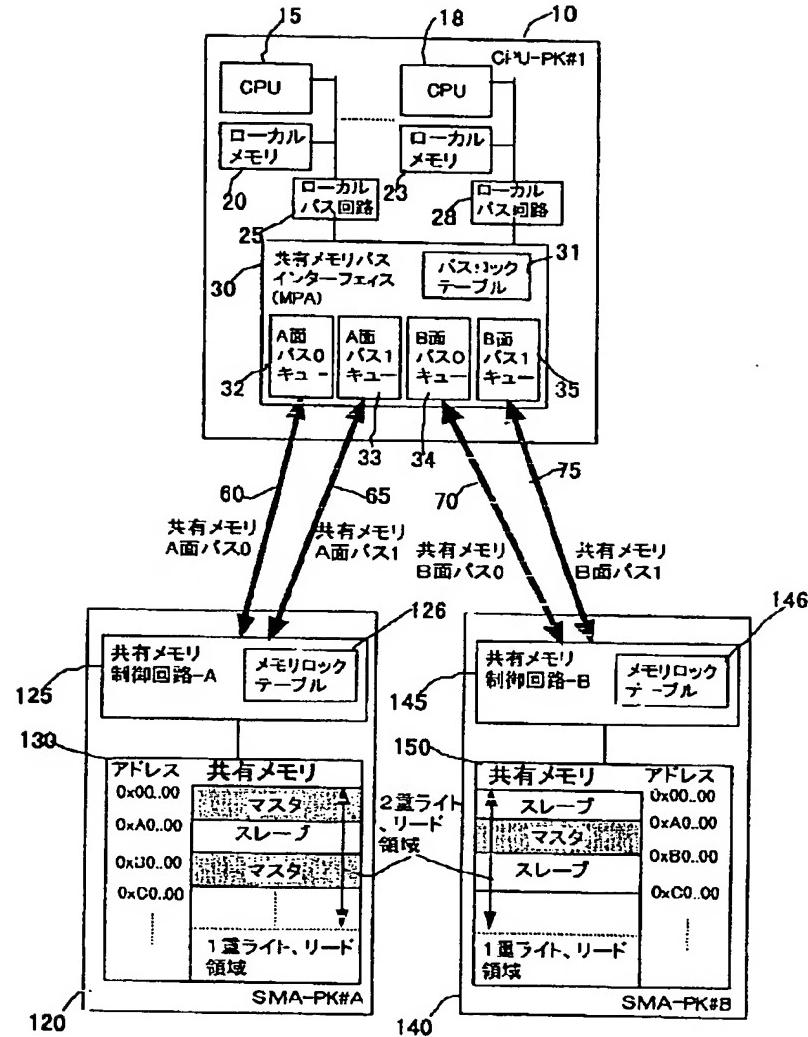
図2



(卓2) 100-267815 (P2000-267815A)

【図3】

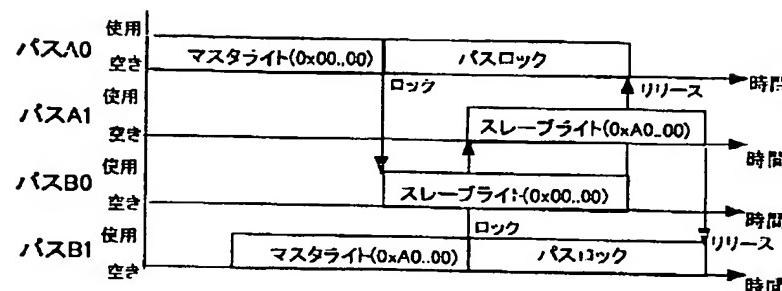
図3



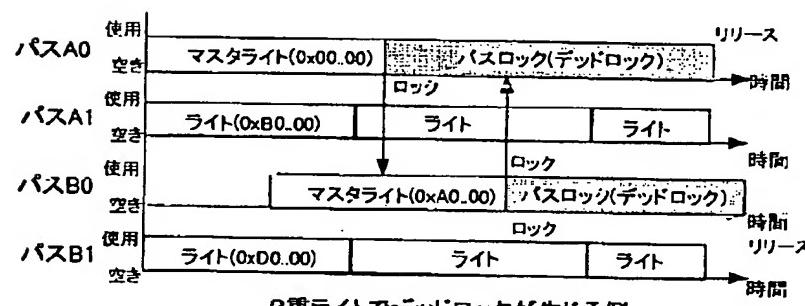
(図 3) 100-267815 (P2000-267815A)

【図4】

図4



2重ライトが効率よく処理できる例



2重ライトでデッドロックが生じる例

(14) 100-267815 (P 2000-267815A)

【図5】

図5

バスロックテーブル

共有メモリ面	使用バス番号	バスキュ一長	キュー内の2重アクセス	処理中のアクセス種類	処理中のロックの有無	処理中のマスター/スレーブの有無
A	60	2	2重アトミックモデファイ	1重リード	無し	無し
A	65	0	無し	1重ライト	無し	無し
B	70	3	無し	2重ライト	有り	マスター
B	/5	2	無し	アトミックモデファイ(リード)	有り	無し

31

【図6】

図6

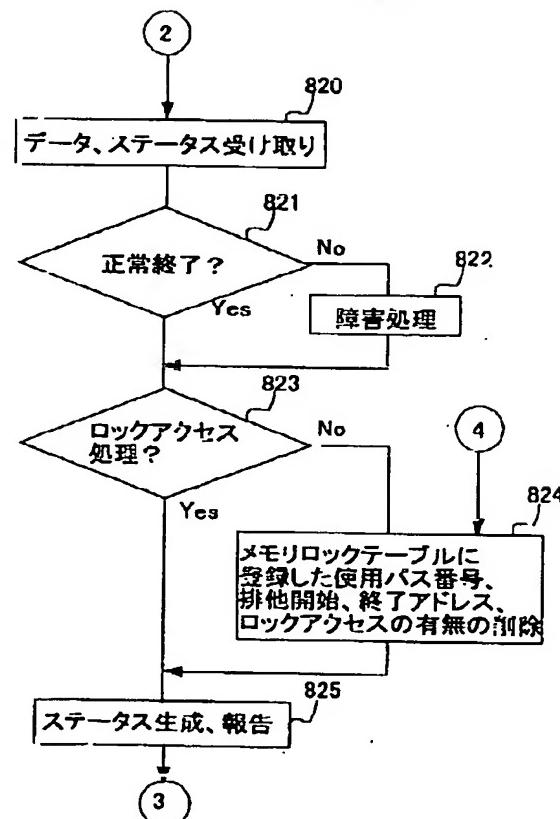
メモリロックテーブル

使用バス番号	排他開始アドレス	排他終了アドレス	ロックアクセスの有無
60	0xA00..07	0XA00..0F	マスター
61	0xB00..FF	0xC00..FF	アトミックモデファイ
62	0xC00..20	0xC00..27	
8	0xD00..30	0xD00..40	アトミックモデファイ

126, 146

【図15】

図15

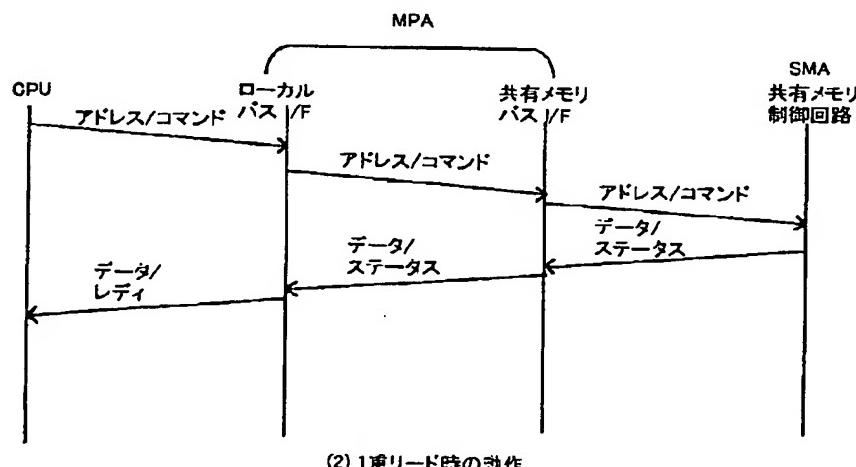
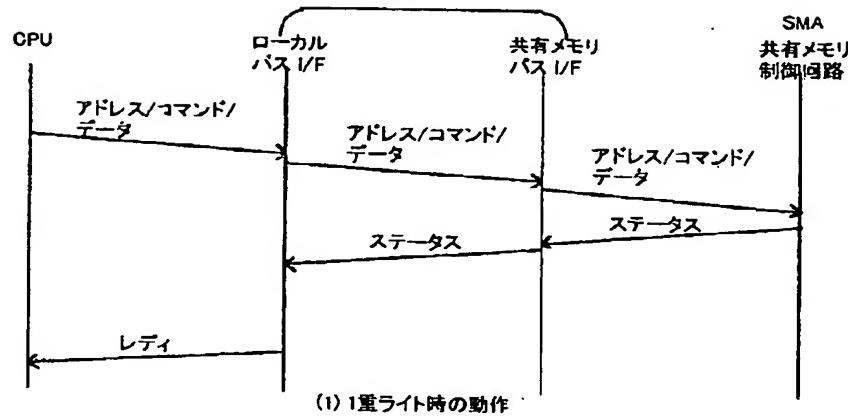


(15) 100-267815 (P 2000-267815A)

【図7】

図7

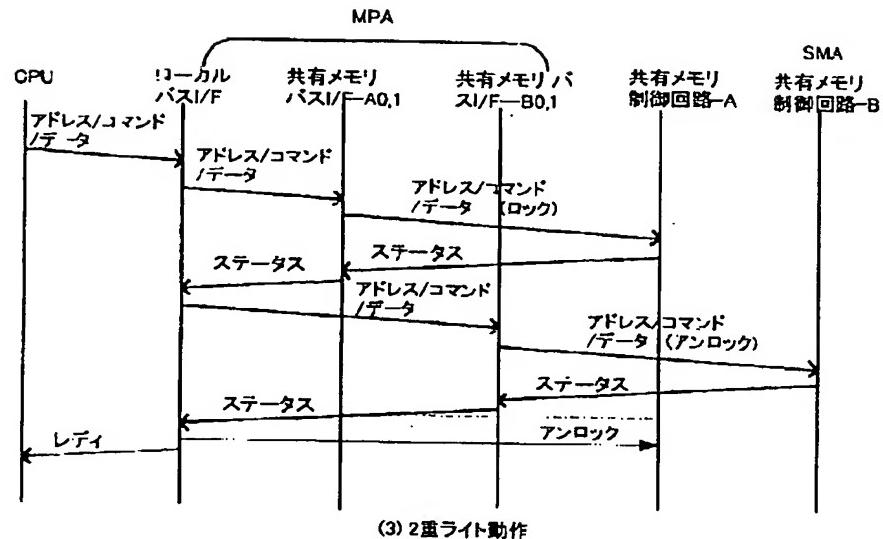
MPA



(専6) 00-267815 (P2000-267815A)

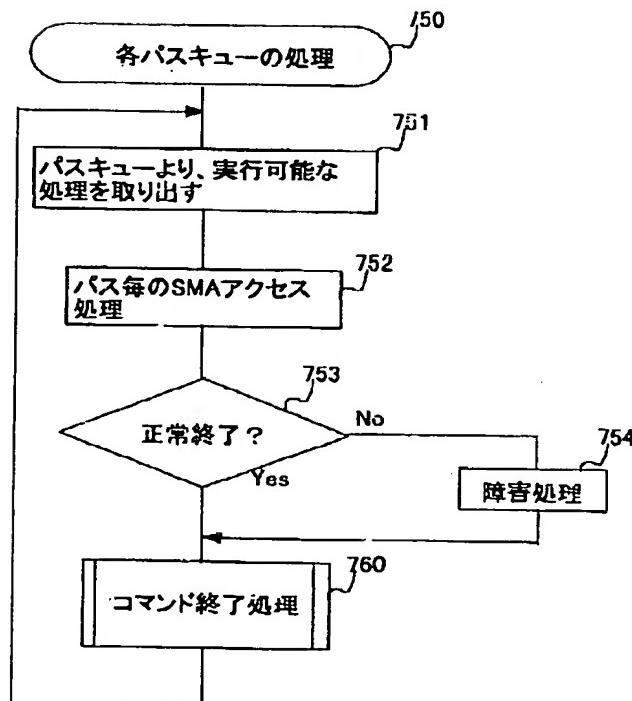
【図8】

図8



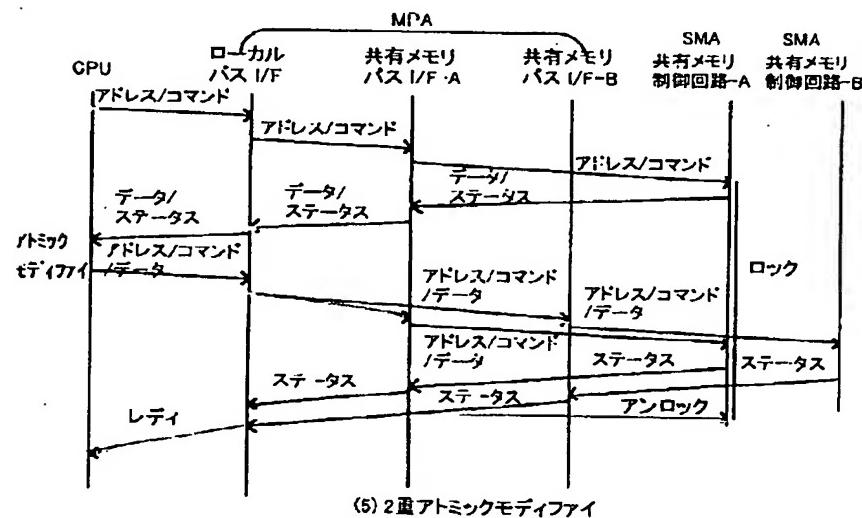
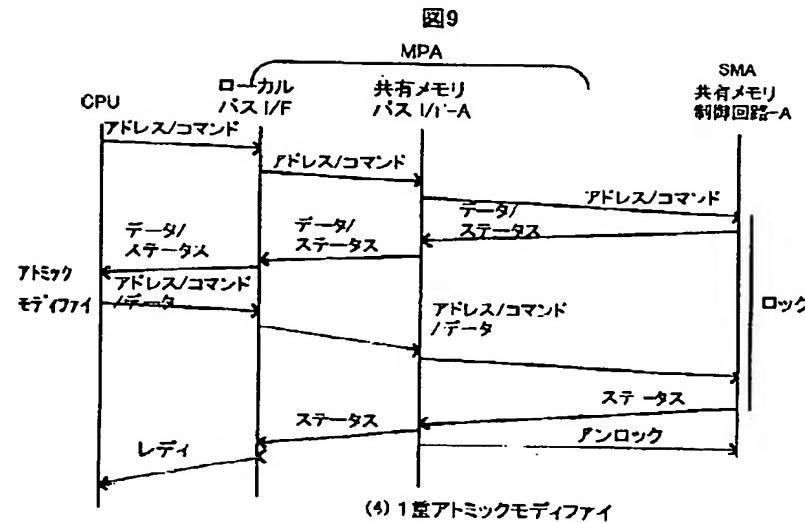
【図12】

図12



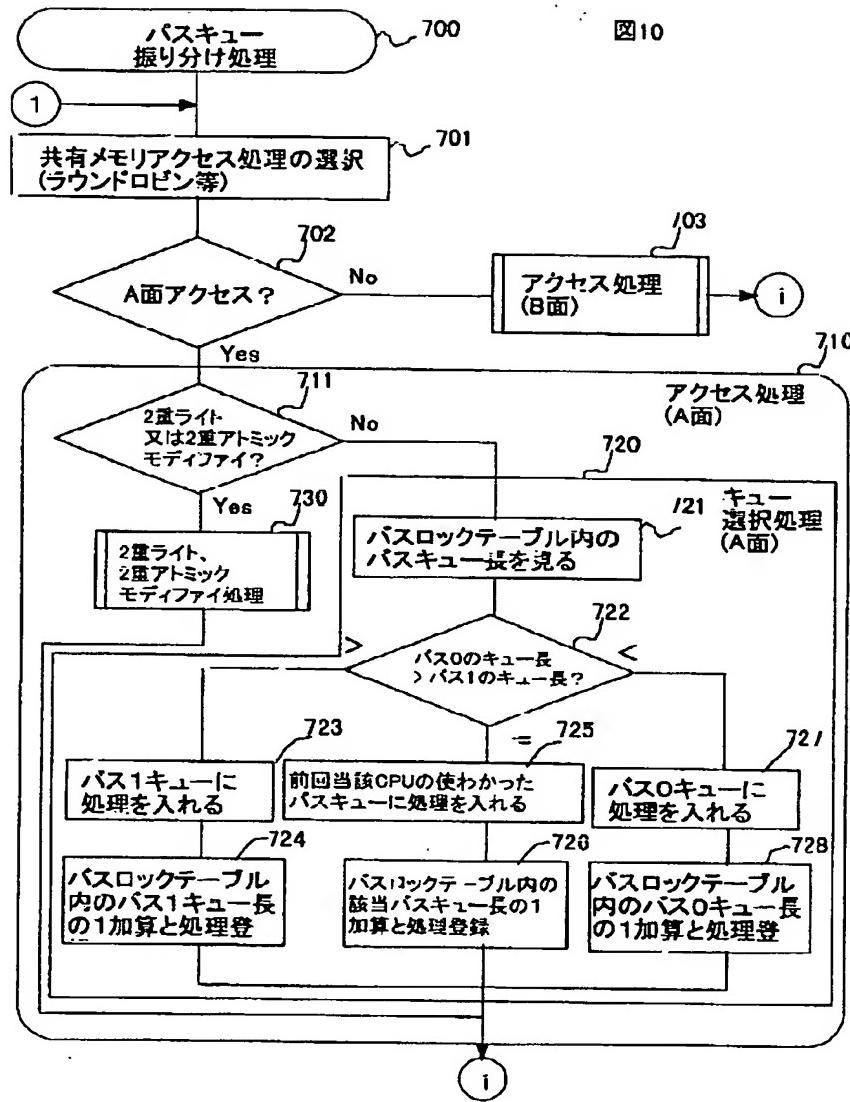
(17) 100-267815 (P2000-267815A)

〔図9〕



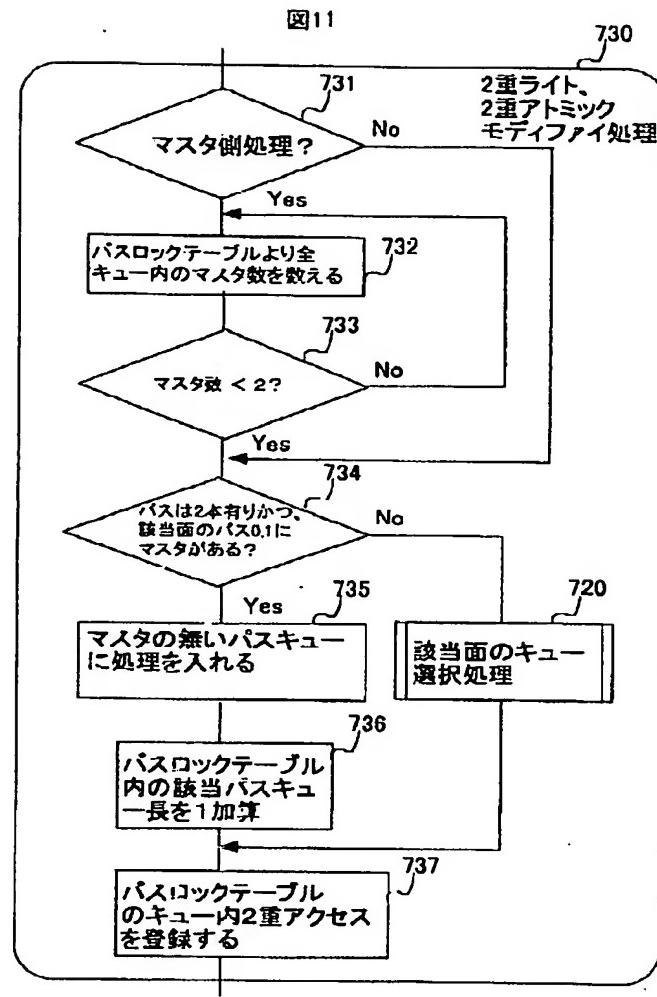
(18) 000-267815 (P2000-267815A)

【図10】



(19) 100-267815 (P2000-267815A)

【図11】



(20) 100-267815 (P2000-267815A)

【図13】

図13  
コマンド終了処理

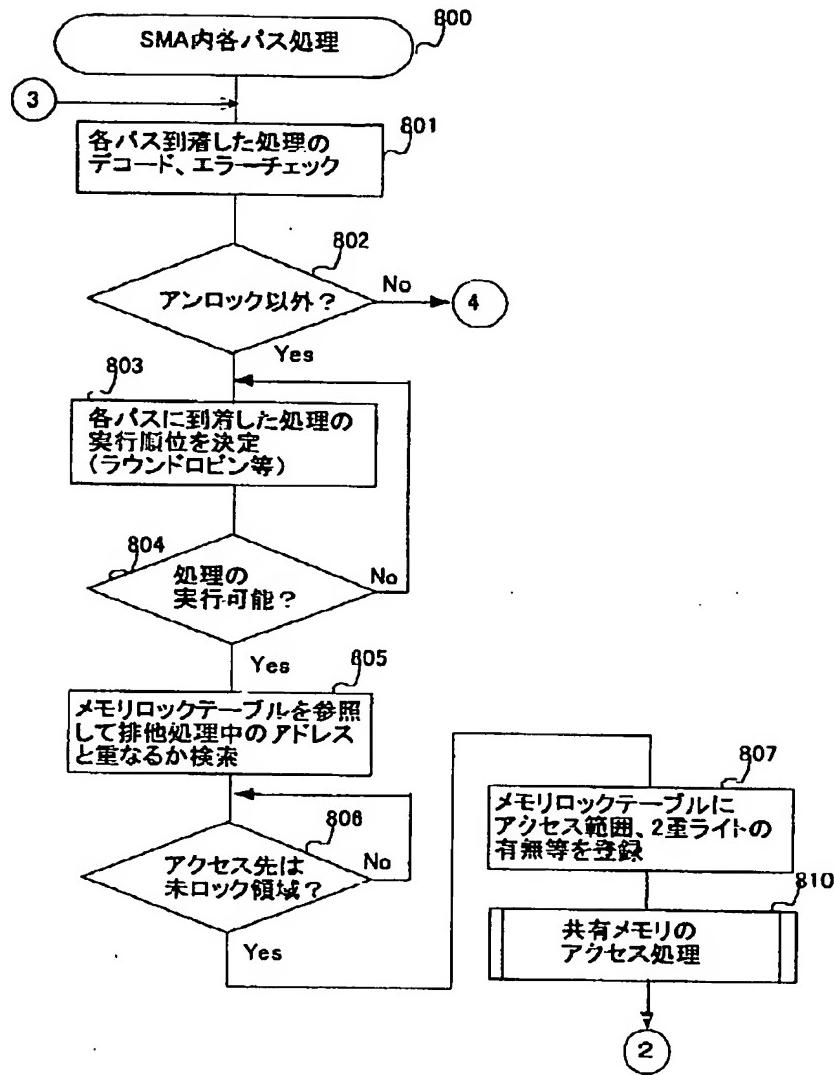
760

処理中の アクセス種	処理中の ロックの有無	処理中の マスタ/ スレーブの有無	次に行う処理
1重リード	無し	無し	バスの開放、 バスロックテーブルに 登録した処理の削除、 データーステータス生成、報告
1重ライト	無し	無し	バスの開放、 バスロックテーブルに 登録した処理の削除、 ステータス生成、報告
2重ライト	有り	マスタ	バスをロックする、 スレーブ側にライトコマンドを発信 スレーブ処理を待つ
2重ライト	有り	スレーブ	マスタ、スレーブバスの開放、 バスロックテーブルに 登録したマスタまたは スレーブ等の処理の削除、 ステータス生成、報告
1重アトミック モディファイ (リード)	有り	無し	バスをロックする、 データ、ステータス生成、報告 バスロックテーブルをライトに変更、 モディファイ後ライトコマンド発行
1重アトミック モディファイ (ライト)	有り	無し	バスの開放、 バスロックテーブルに 登録した処理の削除、 ステータス生成、報告
2重アトミック モディファイ (リード)	有り	無し	バスをロックする、 データ、ステータス生成、報告 バスロックテーブルをライトに変更、 モディファイ後、マスタ、スレーブ側 にライトコマンド発行
2重アトミック モディファイ (ライト)	有り	マスタ	バスをロックする、 スレーブ処理を待つ
2重アトミック モディファイ (ライト)	有り	スレーブ	マスタ、スレーブバスの開放、 バスロックテーブルに 登録したマスタまたは スレーブ等の処理の削除、 ステータス生成、報告

(21) 00-267815 (P2000-267815A)

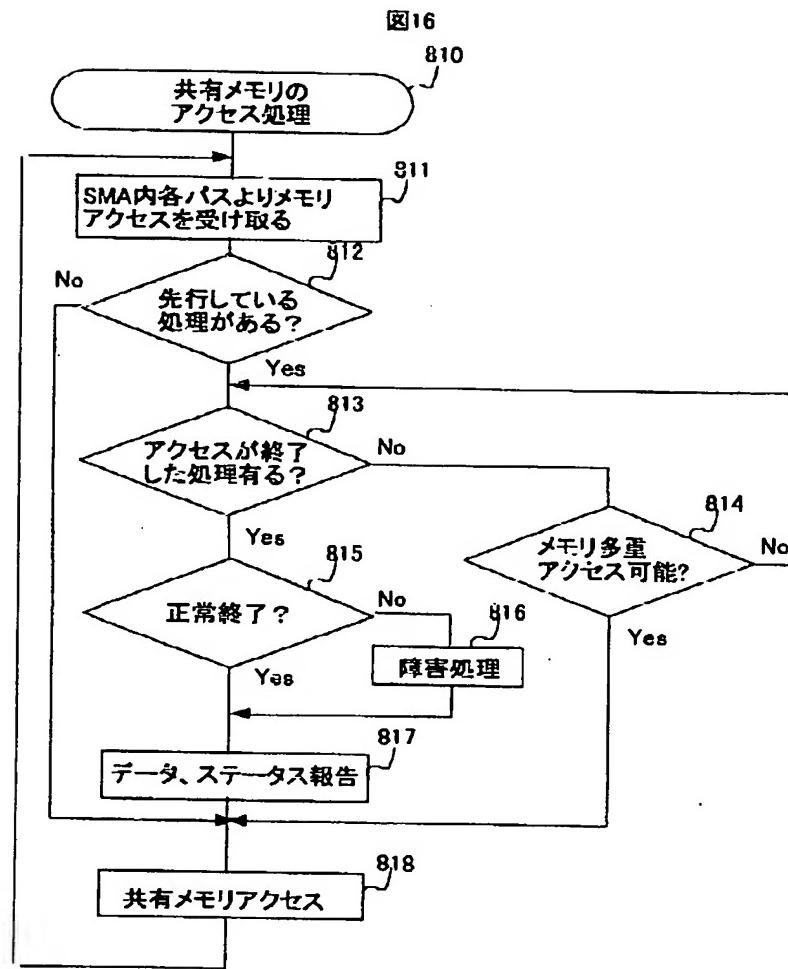
【図14】

図14



(22) 00-267815 (P2000-267815A)

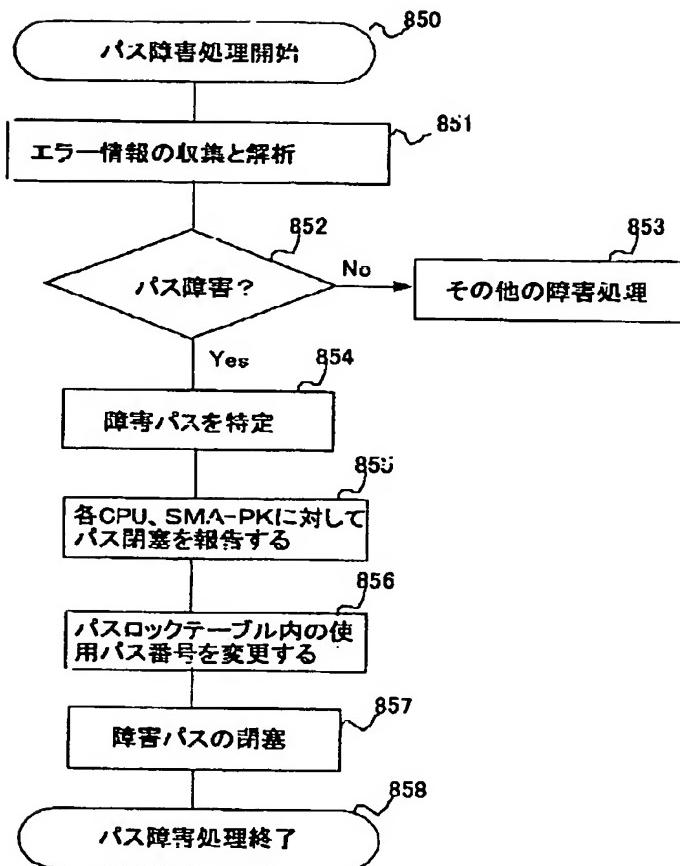
【図16】



(23) 00-267815 (P2000-267815A)

【図17】

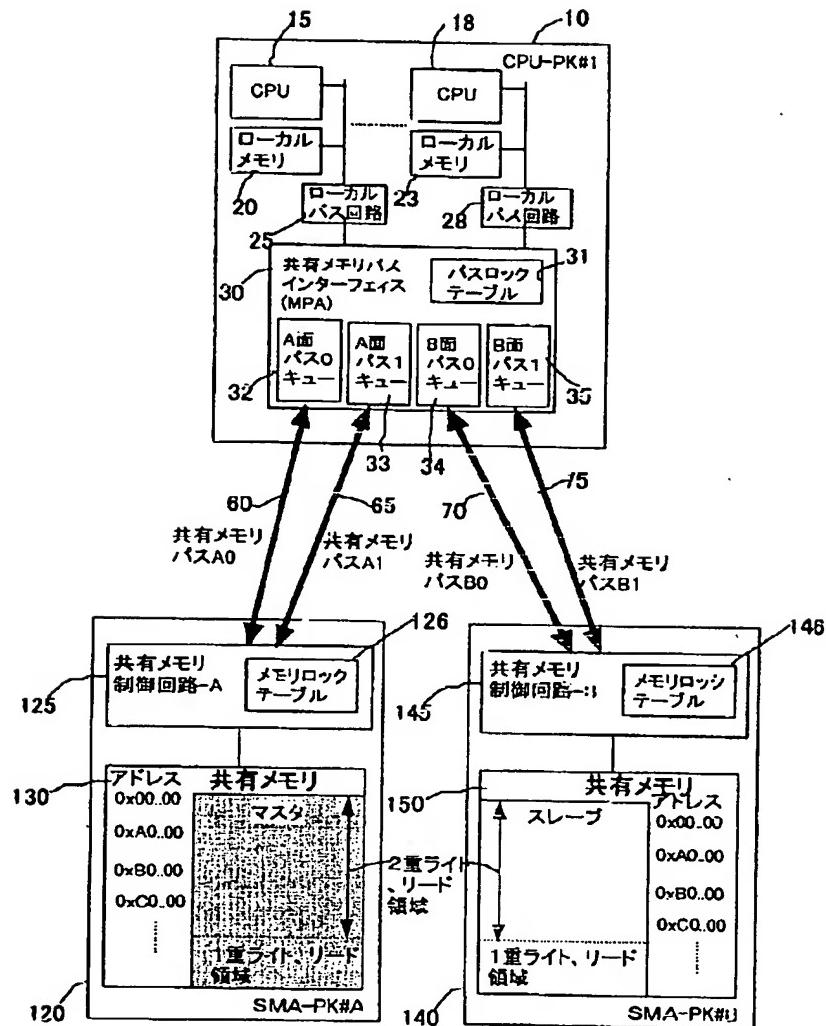
図17



(24) 00-267815 (P2000-267815A)

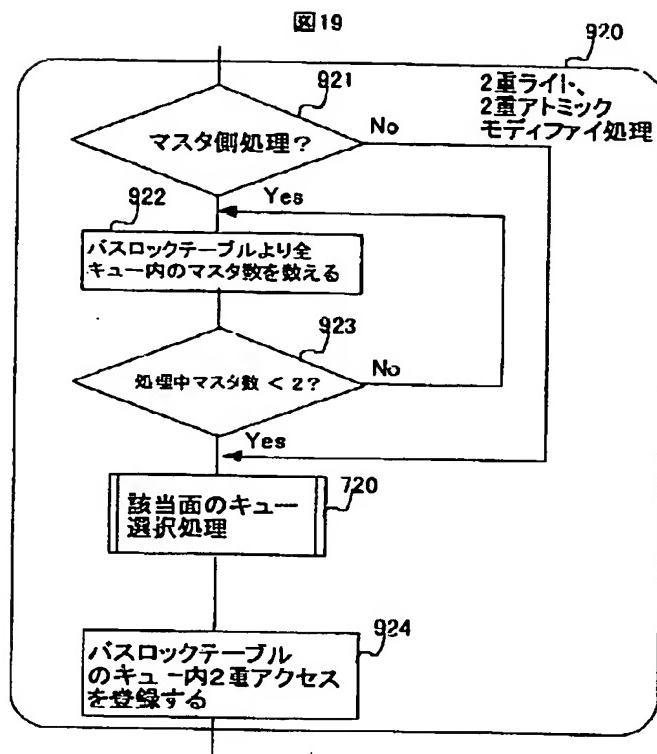
【図18】

図18



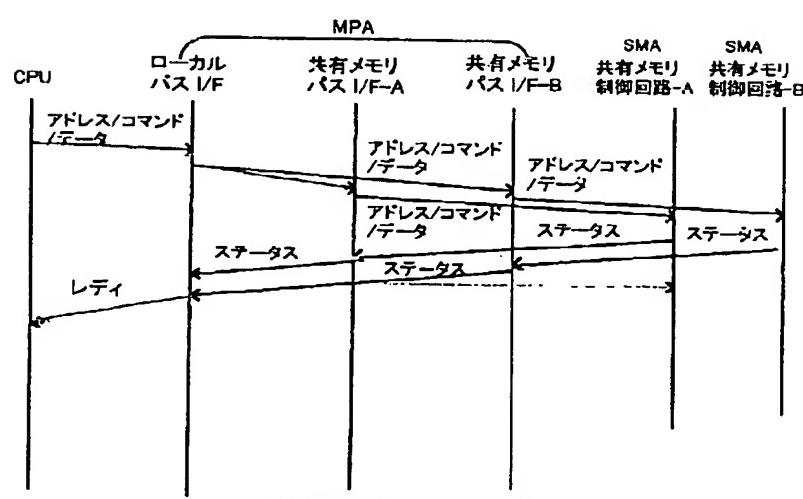
(25) 100-267815 (P2000-267815A)

【図19】



【図20】

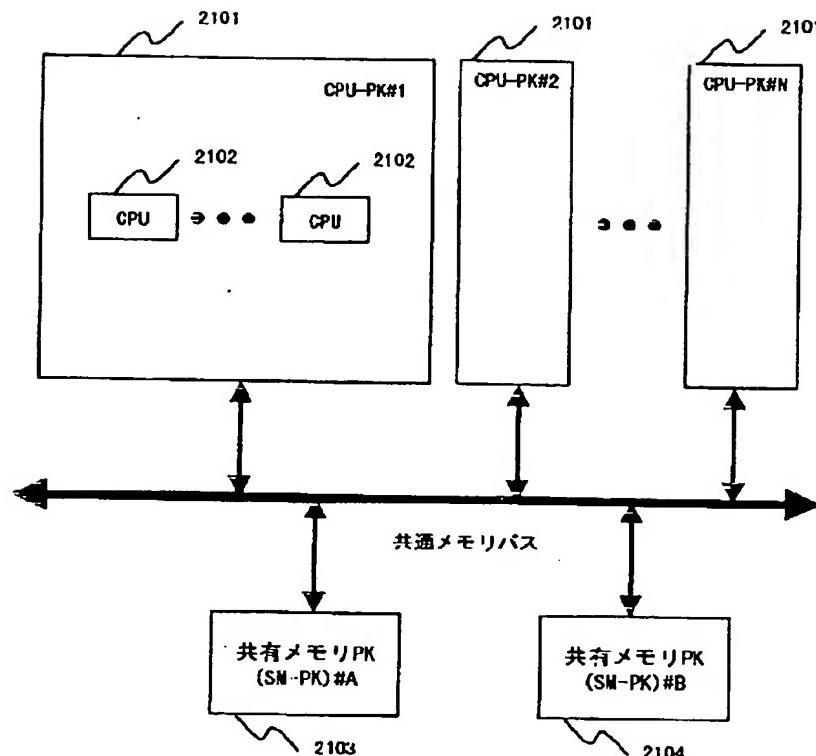
図20



(26) 00-267815 (P 2000-267815A)

【図21】

図21



## フロントページの続き

(72)発明者 田中 淳  
東京都国分寺市東恋ヶ窪一丁目280番地  
株式会社日立製作所中央研究所内  
(72)発明者 藤林 昭  
東京都国分寺市東恋ヶ窪一丁目280番地  
株式会社日立製作所中央研究所内  
(72)発明者 藤本 和久  
東京都国分寺市東恋ヶ窪一丁目280番地  
株式会社日立製作所中央研究所内  
(72)発明者 金井 宏樹  
東京都国分寺市東恋ヶ窪一丁目280番地  
株式会社日立製作所中央研究所内

(72)発明者 箕輪 信幸  
神奈川県小田原市国府津2880番地 株式会  
社日立製作所ストレージシステム事業部内  
(72)発明者 三上 光  
神奈川県横浜市中区尾上町6丁目81番地  
日立ソフトウェアエンジニアリング株式会  
社内  
(72)発明者 浅利 誠  
神奈川県横浜市戸塚区吉田町292番地 株  
式会社日立画像情報システム内  
F ターム(参考) 5B065 BA01 CA12 CA30 CE11 CH13  
EA40 ZA13